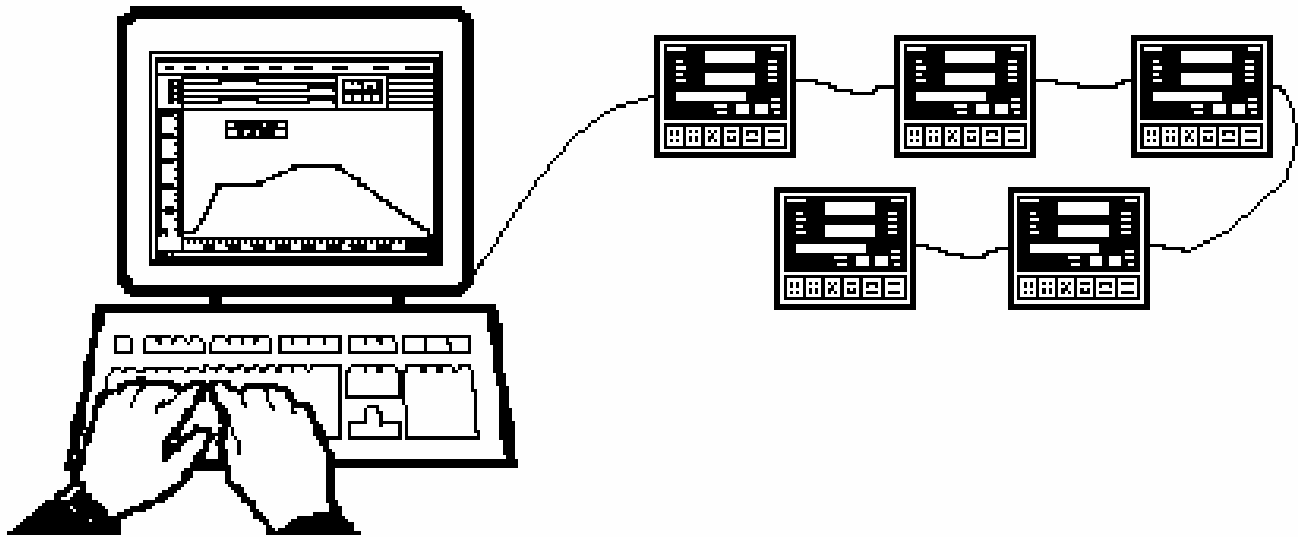


MIC 1460/1462 COMMUNICATION PROTOCOL MANUAL



PREFACE

For high level users who are using a MMI (Man Machine Interface) or SCADA (Supervisory Control And Data Acquisition) program, Section 2 identify all West ASCII communication registers, and in Section 6 identify all MODBUS communication registers which supply all the needed information for interfacing with the instrument's parameters.

For the low level user who will be developing a communication software program, the main portions of this manual will address the message framing and construction, instrument addressing, and all other pertinent details for the applicable protocol. Familiarity with communications and your network is assumed.

The Partlow-West Company reserves the right to modify our products in line with our policy of continuous product improvement. Information contained in this document is subject to change without notice and does not form any part of any contract between the Partlow-West Company and the customer. If you have any suggestions for improvements, or find any errors in this publication, please notify the Partlow-West Company.

MODBUS is a trademark of Modicon, Inc., Industrial Automation Systems.

Table of Contents

| | Page |
|---|-------------|
| SECTION 1.0 - INTRODUCTION | 4 |
| 1.1 RS-485 Communications | 4 |
| 1.2 Available Protocols | 4 |
| 1.3 Communications Enable/Disable | 4 |
| 1.4 Selection of Communications Address | 4 |
| 1.5 Character Transmission | 4 |
| SECTION 2.0 - WEST ASCII PROTOCOL | 5 |
| 2.1 Type 1 Message..... | 6 |
| 2.2 Type 2 Message..... | 6 |
| 2.3 Type 3 Message..... | 7 |
| 2.4 Type 4 Message..... | 7 |
| 2.5 Programmer Parameters and Identifiers..... | 8 |
| 2.5 Programmer Parameters - Functions..... | 9 |
| 2.6 Programmer Commands..... | 10 |
| 2.7 Programmer Status 1..... | 11 |
| 2.8 Current Segment Event Status..... | 11 |
| 2.9 Power Fail Recovery..... | 11 |
| 2.10 Program/Segment Definitions..... | 11 |
| 2.11 Program Scan Table..... | 12 |
| 2.12 Segment Mode..... | 12 |
| 2.13 Programmer Status 2..... | 13 |
| 2.14 User Tag Names..... | 13 |
| 2.15 Controller Parameters and Identifiers..... | 14 |
| 2.16 Controller Scan Table | 15 |
| 2.17 Error Response..... | 17 |
| 2.15 Master Communication Mode..... | 17 |
| SECTION 3.0 - MODBUS RTU PROTOCOL | 18 |
| 3.1 MODBUS Message Framing..... | 18 |
| 3.2 RTU Framing..... | 18 |
| 3.3 RTU Mode..... | 19 |
| 3.4 Contents of the Data Field..... | 19 |
| 3.5 MODBUS Functions Supported..... | 19 |
| 3.6 Message Formats..... | 20 |
| 3.7 Read n Bits - Function Number 01/02..... | 21 |
| 3.8 Read n Words - Function Number 03/04..... | 21 |
| 3.9 Write 1 Bit - Function Number 05..... | 21 |
| 3.10 Write 1 Word - Function Number 06..... | 21 |
| 3.11 Write n Word - Function Number 16..... | 22 |
| 3.12 Loopback Diagnostic Test - Function Number 08..... | 22 |
| 3.13 Error and Exception Responses..... | 22 |
| SECTION 4.0 - PARAMETER VALUE TYPES | 23 |
| 4.1 Bit Parameter Communications..... | 23 |
| 4.2 Single Parameter Value Communications..... | 23 |
| 4.3 Multiple parameter Value Communications..... | 23 |
| SECTION 5.0 - EXAMPLES | 24 |
| 5.1 Read/Write Examples..... | 24 |
| SECTION 6 - MODBUS COMMUNICATION REGISTERS | 25 |
| 6.1 Bit Parameters..... | 25 |
| 6.2 Word Parameters..... | 25 |
| 6.3 Segments Data..... | 26 |

SECTION 1.0 - INTRODUCTION

1.1 RS-485 COMMUNICATION

Communication is at a user selectable rate from the ranges 1200, 2400, 4800 and 9600 baud and the cable used should be suitable for data transfer at this frequency and for the distance required. The transmitters/receivers in the controllers conform to recommendations contained in EIA Standard RS-485.

1.2 AVAILABLE PROTOCOLS

The MIC 1460 software revision **105A 08** and higher , and the MIC 1462 Setpoint Programmer have two communication protocols selectable from the Configuration Mode in the instrument:

- (1) West ASCII
- (2) Modbus RTU

For software Revisions lower then **105A 08**, the only available protocol is the West ASCII protocol.

1.3 COMMUNICATIONS ENABLE/DISABLE

When the communications are enabled (in Controller Define Mode - see Operators manual), the MIC 1460 setpoint programmer's parameters may be adjusted by the master device via the serial communications link. If communications are disabled, the MIC 1460 will not adjust or change any parameters in response to commands received from the master device and will send a negative acknowledgment in response to such commands. Whether communications are enabled or disabled, the controller will return the requested information in response to a Type 2 Interrogation Message (see section 2.2) from the master device (for West ASCII).

1.4 SELECTION OF COMMUNICATIONS ADDRESS

The address for each setpoint programmer is defined in Configuration Mode. For the Modbus protocol, the usable address range is 1 to 247 (per Modbus specifications), even though the instrument can be programmed up to 255. For the West ASCII , the usable address range is 1 to 99.

1.5 CHARACTER TRANSMISSION

For the WEST ASCII protocol the data format is fixed to be even parity, seven data bits and one stop bit.

For the Modbus RTU protocol the data format is 1 start bit, 8 data bits, selectable parity, 1 stop bit if using parity, 2 stop bits if no parity. See Section 3 for further details.

SECTION 2.0 - WEST ASCII PROTOCOL

The protocol assumes half duplex communications. All communication is initiated by the master device. The master sends a command or query to the addressed slave and the slave replies with an acknowledgment of the command or the reply to the query. All messages, in either direction, comprise:

- 1) A Start of Message character - R (Programmer Parameters) or L (Controller Parameters)
- 2) One or two address characters (uniquely defining the slave)
- 3) A parameter/data string, {DATA}
- 4) An End of Message character

Messages from the master may be one of five types:

- Type 1: {S} {N} ??*
- Type 2: {S} {N} {P} {C}* or R {N} {P} {C} *
- Type 3: {S} {N} {P} # {DATA}* or R {N} {P} # {DATA} *
- Type 4: {S} {N} {P} I * or R {N} {P} I *
- Type 5: {S} {N} \ P S S ? *

Where all characters are in ASCII code and:

- {S} is the Start of Message character L (Hex 4C) or R (Hex 52).
L is used for Controller parameters; R is used for Programmer parameters.
- {N} is the slave MIC 1460's address (in the range 1-99); addresses 1-9 may be represented as a single digit (e.g. 7) or in two-digit form, the first digit being zero (e.g. 07).
- {P} is a character which identifies the parameter to be interrogated/modified
(see Table 2-2 and Table 2-3)
- {C} is the command (see 2.2 Type 2 Message)
- # indicates that (DATA) is to follow (Hex 23)
- {DATA} is a string of numeric data in ASCII code (see Table 2-1)
- P is the Program Number
- SS is the Segment number (01 to 16)
- * is the End of Message character (Hex 2A)

No spaces are permitted in messages. Any syntax error in a received message will cause the slave controller to issue no reply and await the Start of Message character.

TABLE 2-1

| {DATA} Element | Sign/Decimal Point Position |
|----------------|-----------------------------|
| {DATA} Content | Sign/Decimal Point Position |
| abcd0 | +abcd |
| abcd1 | +abc.d |
| abcd2 | +ab.cd |
| abcd3 | +a.bcd |
| abcd5 | -abcd |
| abcd6 | -abc.d |
| abcd7 | -ab.cd |
| abcd8 | -a.bcd |

2.1 Type 1 Message

L {N}??*

This message is used by the master device to determine whether the addressed slave Setpoint Programmer is active. The reply from an active slave Setpoint Programmer is:

L (N)?A*

An inactive Setpoint Programmer will give no reply.

2.2 Type 2 Message

L {N} {P} {C} * or R {N} {P} {C} *

This type of message is used by the master device to interrogate or modify a parameter in the addressed Setpoint Programmer. {P} is the parameter (as defined in Table 2-2) and {C} represents the command to be executed, which may be one of the following:

- + (Hex 2B) to increment the value of the parameter defined by {P}
- (Hex 2D) to decrement the value of the parameter defined by {P}
- ? (Hex 3F) to determine the current value of the parameter defined by {P}

The reply from the addressed MIC 1460 is of the form:

L {N} {P} {DATA} A * or R {N} {P} {DATA} A *

where {DATA} is five ASCII-coded digits whose meaning is shown in Table 2-1. The data is the value requested in a query message or the new parameter value after modification. If the action requested by the message from the master device would result in an invalid value for that parameter (either because the requested new value would be outside the permitted range for that parameter or because the parameter is not modifiable), the MIC 1460 replies with a negative acknowledgment:

L {N} {P} {DATA} N * or R {N} {P} {DATA} N *

The {DATA} string in the negative acknowledgment reply will be indeterminate. If the process variable or the deviation is interrogated while the pv is outside the range of the MIC 1460, the reply is:

L {N} {P} < ?? > 0 A * (if over-range) or L {N} {P} < ?? > 5 A * (if under-range)

Scan Tables

A parameter character] in the message from the master device indicates that a "Scan Table" operation is required. This provides a facility for interrogating the values of a group of parameters and status in a single message from the master device. The reply to such a command would be in the form:

L {N} xx aaaaa bbbbbb ccccc ddddd eeeee A * for Controller Scan Table, or
R {N}] 25 P ss VVVVV TTTTT CCCCC SSSSS EEEE A * for a Program Scan Table.

For the Controller Scan Table response, xx is the number of data digits to follow; this is 20 for a single-control output instrument and 25 for a dual-control output instrument. For further information, refer to sections 2.11 and 2.14.

2.3 Type 3 Message

L {N} {P} # {DATA} * or R {N} {P} # {DATA} *

This message type is used by the master to set a parameter to the value specified in {DATA}. The command is not immediately implemented by the slave Setpoint Programmer; the slave will receive this command and will then wait for a Type 4 message (see below). Upon receipt of a Type 3 message, if the {DATA} content and the specified parameter are valid, the slave Setpoint Programmer replies with:

L {N} {P} {DATA} | * or R {N} {P} {DATA} | *

(where I = Hex 49) indicating that the Setpoint Programmer is ready to implement the command. If the parameter specified is invalid or is not modifiable or if the desired value is outside the permitted range for that parameter, the Setpoint Programmer replies with a negative acknowledgment:

L {N} {P} {DATA} N * or R {N} {P} {DATA} N *

2.4 Type 4 Message

L {N} {P} | * or R {N} {P} | *

This is sent by the master device to the addressed slave Setpoint Programmer following a successful Type 3 message transaction with the same Setpoint Programmer. Provided that the {DATA} content and the parameter specified in the preceding Type 3 message are still valid, the slave Setpoint Programmer will then set the parameter to the desired value and will reply with:

L {N} {P} {DATA} A * or R {N} {P} {DATA} A *

where {DATA} is the new value of the parameter. If the new value or parameter specified is invalid, the slave Setpoint Programmer will reply with a negative acknowledgment :

L {N} {P} {DATA} N * or R {N} {P} {DATA} N *

where {DATA} is indeterminate. If the immediately-proceeding message received by the slave Setpoint Programmer was not a Type 3 message, the Type 4 message is ignored.

2.5 PROGRAMMER PARAMETERS (START OF MESSAGE CHARACTER = R)

The Programmer Parameters and their identifier characters are listed in Table 2-2 and their functions are shown in Table 2-3. Unless otherwise stated, the {DATA} element will follow the standard five-digit format and the decimal point position must be correct for the new value to be accepted and for modification to occur.

TABLE 2-2 PROGRAMMER PARAMETER AND IDENTIFIERS

| Identifier | Parameter/Command | Comments | Type |
|------------|------------------------------------|---|------------|
| A | Segment Number | Selects segment in selected program | Read/Write |
| B | Final Setpoint Value | In selected segment | Read/Write |
| C | Segment Time/Ramp Rate | In selected segment | Read/Write |
| G | No. of Cycles Programmed | For selected program | Read/Write |
| H | No. of Cycles Completed | For selected program | Read Only |
| I | Current Segment Number | In currently running/held program | Read Only |
| J | Segment Time Remaining | In currently-running segment | Read Only |
| K | Programmer Commands | Program control/selection | Write Only |
| L | Programmer Status 1 | Summary of programmer parameters | Read Only |
| M | Current Setpoint Value | In currently-running/held program | Read Only |
| N | Current Segment Event Status | Status of each of four Events | Read Only |
| O | Recovery Type (1462 only) | Selects Power Fail recovery Type | Read/Write |
| P | Current Program Number | Indicates program currently running/held | Read Only |
| Q | Delay | Defines delay to start of all programs | Read/Write |
| R | Preset x60 Mode | ON, OFF or not selected (for selected program only) | Read/Write |
| S | Power Failure Recovery | Selects Cold Start or Warm Start recovery | Read/Write |
| T | Program Number | Selects program for comms operations | Read/Write |
| U | User Program Tags | Defines tag name for a program | Read/Write |
| V | Hold Band Value | In selected program | Read/Write |
| W | Hold Time Value | Time in manual hold | Read Only |
| X | Auto-Hold Type | Selects Auto-Hold operation | Read/Write |
| Y | Hold On | Ramps only, Dwells only or both | Read/Write |
| [| External Selection Control | Enables/disables external program selection and/or Run/Hold/Abort control | Read/Write |
| \ | Segment Definitions | Reads/defines details of stated Program/segment | Read/Write |
|] | Program Scan Table | Reads summary of stated program/segment | Read Only |
| - | Segment Mode | Time Mode or Rate Mode | Read/Write |
| ‘ | Programmer Status 2 | Summary of programmer parameters | Read Only |
| ^ | Go To (for 1462 only) | Defines Base mode status at the end of each program | Read/Write |

TABLE 2-3 PROGRAMMER PARAMETERS -FUNCTIONS

| Five-Digit {DATA} Element | | | |
|----------------------------------|--|--------------------------------------|--|
| Parameter | Adjustment Range | Decimal Pos Pt. | Dependent Upon: |
| Segment Number | 0001 to 0016 | 0 | Program Number |
| Final Setpoint Value | Input Range Min - Input Range Max | As per input | Program Number Segment Number |
| Segment Time/ Ramp Rate | Time Mode: 0000 - 9959 (2 right-most digits must not exceed 59) Rate Mode: 0000 - 9999 000n - Join to Program n (n = 1 to 8), 0009 - Repeat Segment, 0010 - End Segment | 2 As per input 5 5 5 | Program Number Segment Number |
| Number of cycles Programmed | 0000 (infinite) - 9999 | 0 | Program Number |
| Number of Cycles Completed | Read Only ; value range 0000 - 9999 and < HH > 0 (numeric limit exceeded) | 0 | Current Program # |
| Current Segment # | Read Only ; value range 0001 - 0016 | 0 | Current Program # |
| Segment Time Remaining | Read Only ; value range 0000 - 9959 (2 right-most digits must not exceed 59) and < HH > 2 (numeric limit exceeded) | 2 | Current Program # Current Segment # |
| Programmer Commands | Write Only; value range 0001 - 0013, 0021 - 0028 (see Section 2.6) | 0 | - |
| Programmer Status 1 | Read Only (see Section 2.7) | N/A | - |
| Current Setpoint Value | Read Only | As per input | - |
| Current Segment Event Status | Read Only (see section 2.8) | N/A | Current Program # Current Segment # |
| Current Program Number | Read Only ; value range 0001 - 0008 | 0 | - |
| Delay (<i>all programs</i>) | 0000 - 9959 (2 right-most digits must not exceed 59) | 2 | - |
| Preset x60 Mode | 0000 = No Preset state 0001 = Preset x60 ON 0002 = Preset x60 OFF | 0 | Program Number |
| Power Fail Recovery | 0000 = Cold Start 0001 = Warm Start (see section 2.9) | 0 | - |
| Program Number | 0001 - 0008 | 0 | - |
| User Tag Names | See section 2.14 | - | Program Number |
| Hold Band | 0000 - Input Span | As per input | Program Number |
| Hold Time | Read Only ; value range 0000 - 9959 (2 right-most digits must not exceed 59) and < HH > 2 (numeric limit exceeded) | 2 | Current Program Number |

| | | | |
|----------------|--|---|----------------|
| Auto Hold Type | 0000 = OFF 0001 = only on PV above SP 0002 = only on PV below SP 0003 = on both | 0 | Program Number |
|----------------|--|---|----------------|

TABLE 2-3 PROGRAMMER PARAMETERS -FUNCTIONS (cont)

| Five-Digit {DATA} Element | | | |
|--|---|------------------|-----------------|
| Parameter | Adjustment Range | Decimal Pos Pt. | Dependent Upon: |
| Hold On | 0000 = ramps and dwells 0001 = ramps only 0002 = dwells only | 0 | Program Number |
| External Selection Control (<i>all programs</i>) | 0000 = Disabled 0001 = program selection only 0002 = Run/Hold/Abort control only 0003 = both | 0 | - |
| Program/Segment Definitions | See section 2.10 | See Section 2.10 | - |
| Program Scan Table | Read Only (See section 2.11) | See Section 2.11 | - |
| Segment Mode | 0000 = Time 0001 = Rate | 0 | - |
| Programmer Status 2 | Read Only (See section 2.13) | N/A | - |

2.6 PROGRAMMER COMMANDS { P } = K

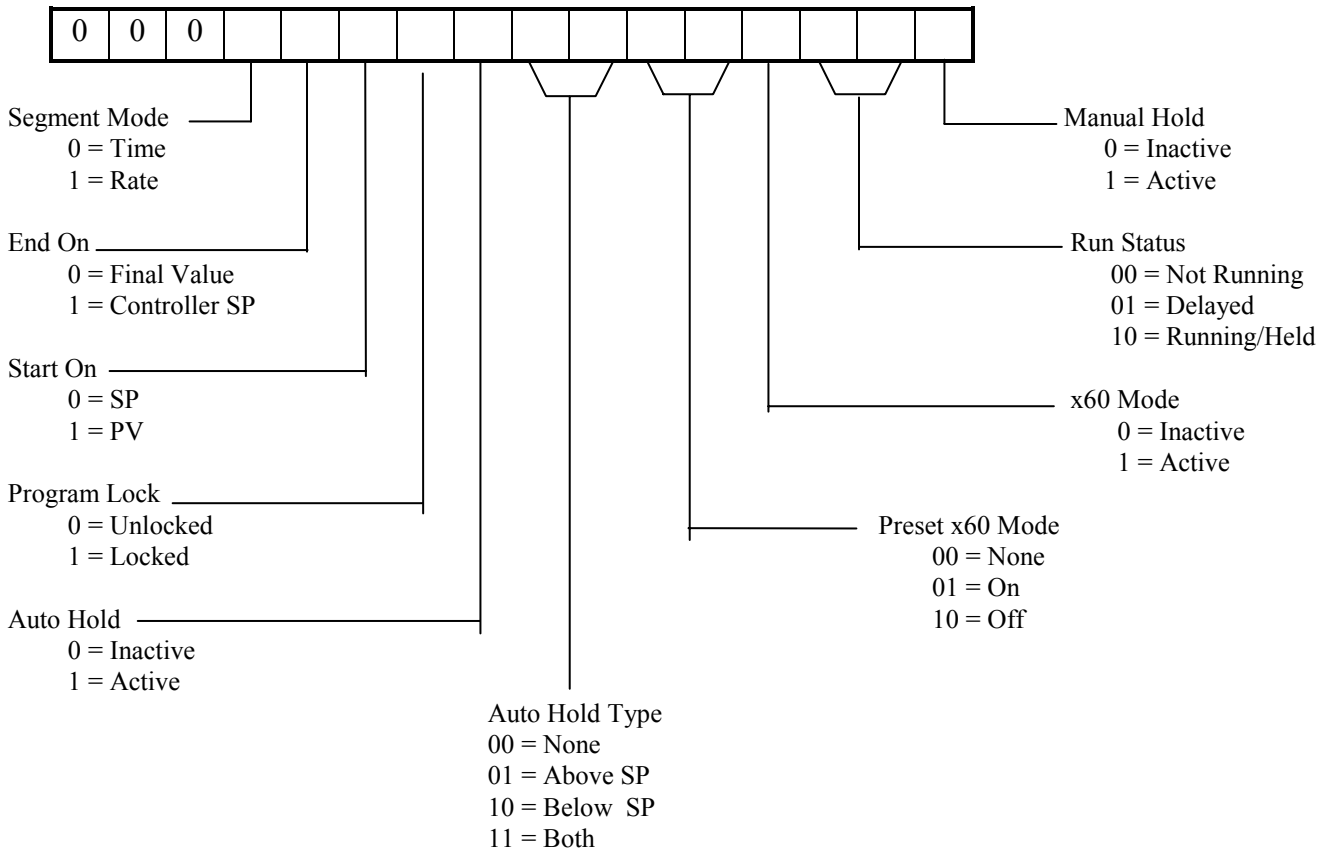
This Write Only parameter is used to convey commands to be implemented by the MIC 1460. In the {DATA} element, the decimal point position digit is set to 0; the remaining four digits are used to specify the command to be executed:

| Value | Command | Value | Command |
|-------|--|-------|--------------------------------|
| 0001 | Run currently-selected program ^{1,2} | 0012 | Enable Program Lock |
| 0002 | Manually Hold currently-running program ¹ | 0013 | Disable Program Lock |
| 0003 | Release Manual Hold ¹ | 0021 | Run Program 1 ^{1,2,3} |
| 0004 | Jump forward one segment ² | 0022 | Run Program 2 ^{1,2,3} |
| 0005 | Abort currently-running program ¹ | 0023 | Run Program 3 ^{1,2,3} |
| 0006 | Start on Controller setpoint | 0024 | Run Program 4 ^{1,2,3} |
| 0007 | Start on process variable | 0025 | Run Program 5 ^{1,2,3} |
| 0008 | End on Final Value | 0026 | Run Program 6 ^{1,2,3} |
| 0009 | End on Controller setpoint | 0027 | Run Program 7 ^{1,2,3} |
| 0010 | Enable x60 Mode | 0028 | Run Program 8 ^{1,2,3} |
| 0011 | Disable x60 Mode | | |

1. If External Run/Hold/Abort Control is active, a negative acknowledgment (NAK) is returned.
2. If Program is in Manual Mode, a negative acknowledgment (NAK) is returned.
3. If External Program Selection is active, a negative acknowledgment (NAK) is returned.

2.7 PROGRAMMER STATUS 1

a This command obtains status information from the MIC 1460. The {DATA} element in the response is five-digit decimal number which when converted into binary form, indicates status as follows:



2.8 CURRENT SEGMENT EVENT STATUS

This Type 2 interrogation message returns the status of the four event outputs for the currently-executed segment (as defined by **Current Segment Number** and **Current Program Number**). The {DATA} element has six digits; the first two are 04 and each of the remaining digits (Event 1 being represented by the left-most digit etc.) may be 1 (active) or 0 (inactive).

2.9 POWER FAIL RECOVERY

This parameter defines the mode of recovery when the power is restored. The {DATA} element decimal point position digit will be set to 0 and the remaining four digits will be set to 0000 (Execute Cold Start - entry into Base Mode with Program Number set as when power failed and Segment Number blank) or 0001. (Execute Warm Start - program resumed from point when power failed). Any attempt to set these digits to a value outside this range will cause a NAK to be returned.

2.10 PROGRAM/SEGMENT DEFINITIONS

This parameter provides a single-message method of setting a given segment in a given program to a defined state and of reading segment details without reference to the **Program Number** and **Segment Number** parameters. It is interrogated using a Type 5 message. The normal response to the interrogation of this form is:

R{N}\14PSSFFFFFFTTTTTEA*

where:

| | |
|-------|---|
| {N} | is the communication address (01 - 99 or 1 -99) |
| 14 | is the character count for the following data |
| P | is the program number |
| SS | is the segment number |
| FFFFF | is the Final Setpoint Value |
| TTTTT | is the Segment Time or Rate (as appropriate) |
| E | is a single hexadecimal character representation of the segment's Event State |

If the Program Number or Segment Number is out of range, a negative response will be returned in the form:

R{N}\14PSSFFFFFFTTTTTEN*

A specified segment may have details written to it by means of a Type 3/5 message sequence.

| | |
|-----------------|--|
| Type 3 Message: | R{N}\#14PSSFFFFFFTTTTTE* |
| Reply: | R{N}\#14PSSFFFFFFTTTTTEI* (positive acknowledgment) |
| | R{N}\#14PSSFFFFFFTTTTTEN* (negative acknowledgment) |

| | |
|-----------------|--|
| Type 5 Message: | Raa\ PSSI* |
| Reply: | Raa\ 14PSSFFFFFFTTTTTEA* (positive acknowledgment) |
| | Raa\ 14PSSFFFFFFTTTTTEN* (negative acknowledgment) |

2.11 PROGRAM SCAN TABLE

This Read Only parameter may be interrogated using a Type 2 message. The normal response is of the form:

R{N}]25PssVVVVVTTTTCCCCSSSSSEEEEA*

where:

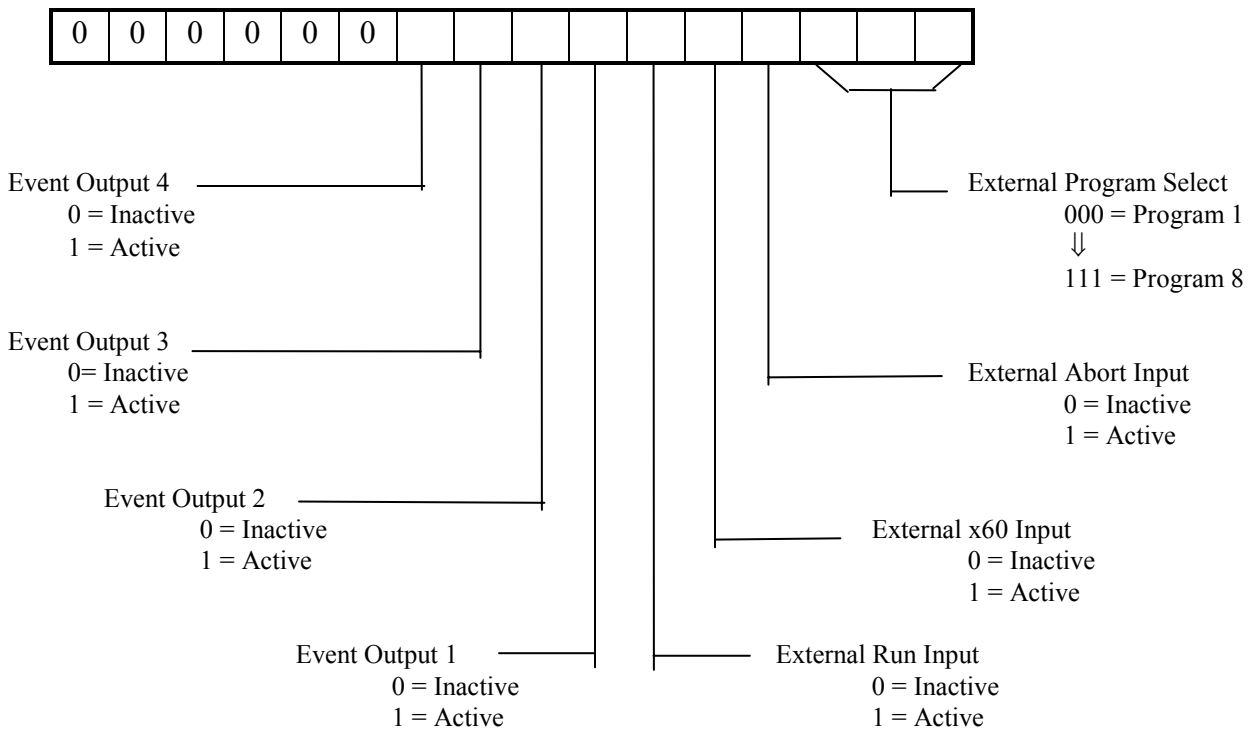
| | |
|-------|---|
| {N} | is the communication address (01 - 99 or 1 -99) |
| P | is the program number |
| ss | is the Segment number |
| VVVVV | is the current setpoint value |
| TTTT | is the Time Remaining/Hold Time (decimal position 2 assumed) |
| CCCC | is the number of cycles completed (0000 to 9999; if higher, will be set to < HH >, signifying "over-range"); decimal point 0 assumed. |
| SSSS | is the Programmer Status 1 (See section 2.7) |
| EEEE | is the Event Status; each digit (Event 1 being represented by the left-most digit etc.) is 1 (active) or 0 (inactive). |

2.12 SEGMENT MODE

This parameter controls the overall programming algorithm of the instrument; Time Mode or Rate Mode. Changing this parameter will cause all the other Program Definitions to be set to default values, which takes in excess of one second to complete. Software which writes to this parameter must therefore take into account this time before expecting a reply.

2.13 PROGRAMMER STATUS 2

This Read Only parameter applies to the Event Output and Program Control Input options. It is interrogated by a Type 2 message only and the response is the normal five-digit {DATA} format which contains a decimal representation (decimal point position is always set to 0) of an eight-bit binary number whose bits have the following significance:



2.14 USER TAG NAMES

Tag names may use all the alphanumeric characters, spaces and standard punctuation marks. They must consist of eight characters, with blanks being represented by space characters.

The interrogation message is the Type 2 message:

Query: R{N}U ? *
 Reply: R{N}U 08 CCCCCCCC A*

The tag names can be written with the usual Type 3/Type 4 message transaction:

Type 3 Message: R{N}U # 08 CCCCCCCC *
 Reply: R{N}U 08 CCCCCCCC I* (positive acknowledgment)
 R{N}U 08 CCCCCCCC N* (negative acknowledgment)

Type 4 Message: R{N}U I *
 Reply: R{N}U 08 CCCCCCCC A*

2.15 CONTROLLER PARAMETERS (Start of Message character = L)

The controller parameter and their identifier characters are listed in Table 2-5 and their functions are listed in Table 2-6.

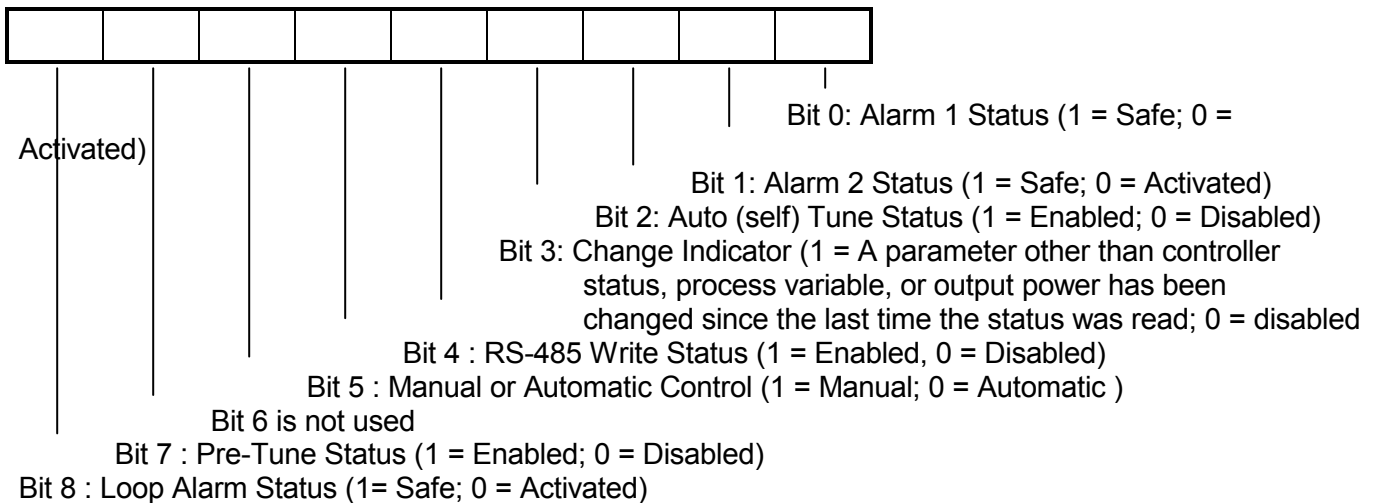
TABLE 2-5 CONTROLLER PARAMETERS AND IDENTIFIERS

| Identifier | Parameter/Command | Comments | Type |
|-------------------|---|--|-------------|
| A | Setpoint High Limit | Maximum setpoint value | Read/Write |
| B | Output 1 Power Limit | Safety power limit | Read/Write |
| C | Alarm 1 Value | Alarm 1 active at this level | Read/Write |
| D | Rate ¹ | Derivative Time Constant value | Read/Write |
| E | Alarm 2 Value | Alarm 2 active at this level | Read/Write |
| F | ON/ OFF Differential Value | Switching Hysteresis for ON/OFF control | Read/Write |
| G | Scale Range Maximum | Upper limit of scaled input range | Read/Write |
| H | Scale Range Minimum | Lower limit of scaled input range | Read/Write |
| I | Reset Value ¹ or Loop Alarm Time | Integral Time Constant value or (for ON/OFF control with Loop Alarm enabled) Loop Alarm value | Read/Write |
| J | Manual Reset | Bias value | Read/Write |
| K | Overlap/Deadband | Portion of PB1 + PB2 over which both outputs are active | Read/Write |
| L | Controller Status ² | Status information (see notes on this table) | Read Only |
| M | Process Variable Value | Universal input | Read Only |
| N | Output 1 Cycle Time | Output 1 Relay cycle time | Read/Write |
| O | Output 2 Cycle Time | Output 2 Relay cycle time | Read/Write |
| P | Proportional Band 1 Value ¹ | Proportion of input span over which Output 1 level is proportional to the PV | Read/Write |
| Q | Scale Range Decimal Point Pos. | Decimal point position | Read/Write |
| S | Setpoint Value | - | Read/Write |
| T | Setpoint Low Limit | Minimum value of setpoint | Read/Write |
| U | Proportional Band 2 Value ¹ | Proportion of input span over which Output 2 level is proportional to the PV | Read/Write |
| V | Deviation Value | PV - SP | Read Only |
| X | Outputs (for 1462 only) | Sets output status in Base mode | Read/Write |
| W | Output Power | Output power level | Read Only |
| Z | Setpoint Programmer Commands ³ | See notes on this table | Write Only |

| | | | |
|---|----------------------------------|--|------------|
| [| Recorder Output Scale Maximum | Max. scale value for recorder output | Read/Write |
| \ | Recorder Output Scale Minimum | Min scale value for recorder output | Read/Write |
|] | Scan Table | Reads main control parameters | Read Only |
| ^ | Setpoint Ramp Rate | - | Read/Write |
| m | Input Filter Time Constant Value | Digital Filter | Read/Write |
| v | Process Variable Offset Value | Modified PV = Actual Process Value minus the PV Offset | Read/Write |
| ^ | SP Ramp Rate (1462 only) | Setpoint Ramp Rate | Read/Write |

NOTES ON TABLE 2-5

1. These parameters can not be modified while either the Pre-Tune or the Auto (Self)-Tune is activated.
2. The Controller Status byte has the following format:



3. Only Type 3 or Type 4 messages are allowed with this parameter. In the Type 3 message, the {DATA} field must be one of the eight five-digit numbers. The reply from the MIC 1460 also contains the {DATA} field with the same content. When the master device issues the Type 4 message, the MIC 1460 responds with the same {DATA} field content. The commands corresponding to the {DATA} field value are:

- 00010 = Active Manual Control
- 00020 = Active Automatic Control
- 00030 = Active Auto (Self) -Tune facility
- 00040 = De-active Auto (Self) -Tune facility
- 00050 = Request Pre-Tune (see note below)
- 00060 = Abort Pre-Tune
- 00130 = Active Loop Alarm
- 00140 = De-active Loop Alarm

*NOTE: The MIC 1460 will go into Pre-Tune mode only if the process variable is at least 5% of input span from the setpoint.

TABLE 2-6 CONTROLLER PARAMETERS -FUNCTIONS

Five-Digit {DATA} Element

| Parameter | Adjustment Range | Decimal Pt Pos. | Default Value |
|--------------------------------------|---|-----------------|----------------|
| INPUT PARAMETERS: | | | |
| Process Variable (PV) ² | None - Read Only | As per input | Not applicable |
| Process Variable Offset ³ | Modified PV limited by Scale Range Max. and Scale Range Min. | As per input | 0 |
| Scale Range Maximum ¹ | | As per input | 1000 |
| Scale Range Minimum ¹ | | As per input | 0000 |
| Scale Range | | | |
| Decimal Point Position ¹ | 0 = xxxx 2= xx.xx 1 = xxx.x 3 = x.xxx | 0 | 1 |
| Filter Time Constant | 0.0 seconds to 100.0 seconds | 1 | 2.0 |

TABLE 2-6 CONTROLLER PARAMETERS -FUNCTIONS (cont)

| Five-Digit {DATA} Element | | | |
|-----------------------------------|--|---------------------------------|--------------------------|
| Parameter | Adjustment Range | Decimal Pt Pos. | Default Value |
| OUTPUT PARAMETERS: | | | |
| Power Output Value ⁴ | 0 to 100% for 1 output; -100% to +100% for 2 outputs | 0 | - |
| Output 1 Power Limit ⁸ | 0% to 100% | 0 | 100% |
| Output 1 Cycle Time | Powers of 2 in the range 0.5 secs. to 512 secs. (.5, 1, 2, etc.) | 0 or 1 | 32 secs. |
| Output 2 Cycle Time | Powers of 2 in the range 0.5 secs. to 512 secs. (.5, 1, 2, etc.) | 0 or 1 | 32 secs. |
| Recorder Output Scale Max. | -1999 to 9999 | As per input | Scale Range Max. |
| Recorder Output Scale Min. | -1999 to 9999 | As per input | Scale Range Min. |
| SETPOINT PARAMETERS | | | |
| Setpoint (SP) Value | Setpoint Low Limit to Setpoint (SP) High Limit | As per input | - |
| Setpoint (SP) High Limit | Current SP to Input Range Max. | As per input | Input Range Max. |
| Setpoint (SP) Low Limit | Current SP to Input Range Min. | As per input | Input Range Min. |
| ALARM PARAMETERS | | | |
| Alarm 1 Value | Depends upon type ⁵ | As per input | As per type ⁵ |
| Alarm 2 Value | Depends upon type ⁵ | As per input | As per type ⁵ |
| TUNING PARAMETERS | | | |
| Rate ⁸ | 00 secs. to 99 mins. 59 secs. ⁶ | 2 | 1 min. 15 secs |
| Reset ⁸ | 1 sec. to 99 mins 59 secs. ⁶ | 2 | 5 mins 00 secs. |
| Manual Reset ⁸ | 0% to 100% for 1 output; -100% to +100% for 2 outputs | 0 (positive) or 5 (negative) | 25% |
| ON/OFF Differential | 0.1% to 10.0% of input span | 1 | 0.5% |
| Overlap/Deadband ^{7, 8} | -20% to +20% of PB1 + PB2 Negative = Deadband Positive = Overlap | 0 (positive) or 5 (negative) | 0% |
| Proportional Band 1 (PB1) | 0.0% to 999.9% of input span | 1 | 10.0% |

| | | | |
|---|------------------------------|--------------|----------------|
| Proportional Band 2 (PB2) ^{7,8} | 0.0% to 999.9% of input span | 1 | 10.0% |
| STATUS PARAMETERS: | | | |
| Controller Status (see Table 2-5) | Read Only -not adjustable | N/A | Not applicable |
| Arithmetic Deviation | Read Only -not adjustable | As per input | Not applicable |
| Controller Scan Tables (see section 2.16) | Read Only -not adjustable | N/A | Not applicable |

Notes:

1. Applicable to DC linear inputs only.
2. If the process variable is out of range, the {DATA} byte in the reply will be <?? > 0 (over-range) or <?? > 5 (under-range).
3. The PV Offset parameter value should be selected with care, using a high PV Offset could lead to displayed process variable value bearing no meaningful relationship to the actual PV value.
4. If Manual Control is not selected, this is a “Read Only” parameter (i.e. can not be adjusted).
5. The default settings for the different alarm types are as follows:

| Alarm Type | Adjustment Range | Default Setting |
|--------------------|--------------------------------------|------------------------|
| Process High Alarm | Scale Range Min. to Scale Range Min. | Scale Range Max. |
| Process Low Alarm | Scale Range Min. to Scale Range Min. | Scale Range Min. |
| Band Alarm: | 0 to input span from setpoint | 5 input units |
| Deviation Alarm: | ± input span from setpoint | 5 input units |

6. {DATA} element is in the form mm.ss (mm = minutes, ss = seconds).
7. Applicable only if Output 2 is fitted.
8. Not operative if Proportional Band = 0.

2.16 CONTROLLER SCAN TABLE

The Controller Scan Table operation takes the form of a Type 2 interrogation command which accesses a set of information (held in the {DATA} element in the response). The response would be in the form:

L{N}] xx aaaaa bbbbb ccccc ddddd eeeee A *

where xx is the number of data digits in the {DATA} element to follow; this is 20 for a single-control-output instrument and 25 for a dual-control-output instrument. These digits are described in Table 2-1 and may comprise:

| | |
|-------|--|
| aaaaa | The current setpoint value |
| bbbbb | The current process variable value |
| ccccc | The current value of Output 1 Power (0 -100%) |
| ddddd | The current value of Output 1 Power (0 -100%) |
| eeeee | The Controller Status (see Note 2 on Table X-X). |

2.17 ERROR RESPONSE

The circumstance under which a message received from the master device is ignored are:

- Parity error detected
- Syntax error detected
- Time-out elapsed
- Receipt of a Type 4 message without a preceding Type 3 command message.

Negative acknowledgments will be returned if, in spite of the received message being notationally correct, the MIC 1460 can not supply the requested information or perform the requested operation. The {DATA} element of a negative acknowledgment will be indeterminate.

2.18 MASTER COMMUNICATIONS MODE

When the MIC 1460 is set to operate as a master device over an RS-485 communication link, it uses the protocol which was selected in the Configuration Mode (West ASCII or Modbus RTU) to send the current setpoint value to each slave instrument, connected and powered up. The MIC 1460 Setpoint Programmer will detect all addresses which have no slave instrument and will automatically skip to the next address; the MIC 1460 periodically interrogates “empty” addresses to determine whether or not they have become active. The address-to-address scan rate is five per second at 4800 baud and ten per second at 9600 baud.

***NOTE:** Each slave instrument **MUST** be of the same protocol type same type parity as the master.

SECTION 3.0 - MODBUS RTU PROTOCOL

The remainder of this document specifies the MODBUS communications protocol as implemented on applicable Partlow instruments. Source material for this manual is the Modicon MODBUS Protocol Reference Guide PI-MBUS-300, Rev. F.

For the high level user who is using a MMI (Man Machine Interface) program, all you will need to use is the Section 6 to identify all the instruments register assignments and data types.

For the low level user developing a communication software program, this manual will address the message framing and construction, instrument addressing, and all other pertinent information.

This manual does not try to be a complete guide to the MODBUS protocol, but will show how to structure a message that the Partlow instruments will recognize, how to request access to another device, and how errors will be detected and reported.

3.1 MODBUS MESSAGE FRAMING

There are two serial transmission modes for the MODBUS protocol, ASCII or RTU (Remote Transmission Unit) framing. The Partlow instruments use the **RTU** framing method **only**. The MODBUS message is placed by the transmitting device into a frame that has a known beginning and ending point. This allows receiving devices to begin at the start of the message, read the address portion and determine which device is addressed, and know when the message is completed. Partial messages can be detected and errors can be set as a result.

3.2 RTU FRAMING

In RTU framing mode, the message starts with a silent interval of at least 3.5 character times. This is most easily implemented as a multiple of character times at the baud rate that is going to be used in the network. The first field then transmitted is the device address. Binary data is used for transmission for all fields. Network devices monitor the network bus continuously, including the “silent” interval. When the first field is received, each device decodes it to find out if it is being addressed.

Following the last transmitted character, a similar ‘silent’ interval of at least 3.5 character times marks the end of the message. A new message can then begin after this interval. The entire message frame must be transmitted as a continuous stream. If a ‘silent’ interval of more than 1.5 character times occurs before completion of the frame, the receiving device discards the incomplete message and assumes the next byte will be the address field of a new message. If a new message begins earlier than 3.5 character times following a previous message, the receiving device will consider it a continuation of the previous message. This will cause an error, as the value in the final CRC field will not be valid for the combined messages. A typical message frame is shown Figure 1 below.

| START | ADDRESS | FUNCTION | DATA | CRC CHECK | END |
|-------------|---------|----------|------------|-----------|-------------|
| T1-T2-T3-T4 | 8 BITS | 8 BITS | n X 8 BITS | 16 BITS | T1-T2-T3-T4 |

Figure 1 : Typical message frame

Note: All bytes are a total of 11 bits long, which include 1 start bit, 8 data bits, 1 parity bit (if used), and 1 or 2 stop bits (depends if a parity bit is used). See Bits per Byte on next page.

3.3 RTU MODE

The main advantage of the RTU mode is that its greater character density allows better data throughput than ASCII for the same baud rate. Each message must be transmitted in a continuous stream. The format for each byte in RTU mode is:

Coding System: 8-bit binary

- Bits per Byte: 1 start bit
- 8 data bits, least significant bit sent first
- 1 bit for even/odd parity; no bit for no parity
- 1 stop bit if parity is used; 2 bits if no parity

Baud Rate: 1200, 2400, 4800, and 9600

Error Check Field: Cyclical Redundancy Check (CRC)

3.4 CONTENTS OF THE DATA FIELD

The data field of messages sent from a master to a slave device contains additional information which the slave must use to take the action defined by the function code. This can include items like register addresses, the quantity of items to be handled, and the count of actual data bytes in the field.

For example, if the master requests a slave to read a group of holding registers (function code 03), the data field specifies the starting register and how many registers are to be read. If the master writes to a group of registers in the slave (function code 16), the data field specifies the starting register, how many registers to write, the count of data bytes to follow in the data field, and the data to be written in the registers. If no error occurs, the data field of a response from a slave to a master contains the data requested. If an error occurs, the field contains an exception code that the master application can use to determine the next action to be taken.

3.5 MODBUS FUNCTION SUPPORTED

The following is a list the functions to be supported. The JBUS names are listed first, where such an equivalence exists, as these more closely represent the actual operations. The original Gould MODBUS function names are listed for reference. The MODBUS Function number follows the names. In some cases two function numbers will be supported, as they could be used interchangeably with respect to our unit. Please refer to Figure 2 below for the MODBUS functions that are supported.

| | JBUS | MODBUS | FUNCTION NUMBER |
|---|---------------|--|------------------------|
| A | Read n Bits | Read Coil Status Read Input Status | 01 02 |
| B | Read n Words | Read Holding Registers Read Input Registers | 03 04 |
| C | Write 1 Bit | Force Single Coil | 05 |
| D | Write 1 Word | Preset Single Register | 06 |
| E | Loopback Test | Loopback Diagnostic Test | 08 |
| F | Write n Words | Preset Multiple Registers | 16 |

Figure 2 : MODBUS Functions that are supported

The instrument will identify itself in reply to a Read Holding Register message which inquires the value of parameter 121 & 122, as specified in the CNOMO documentation, and the Modbus Function 17 (Report Slave ID) will not be supported.

3.6 MESSAGE FORMATS

The first character of each message is an instrument address. The valid range of such an address is 0 to 247. The second character is always the function number. The contents of the remainder of the message depends upon the function number. The maximum length of a message is 256 bytes.

In most cases the instrument is required to reply by echoing the address and function number, together with an echo of all or part of the message received (in the case of a request to write a value or carry out a command) or the information requested (in the case of a read parameter operation). Broadcast Messages (to which the instrument responds by taking some action, without sending back a reply) are supported at instrument address zero. Commands which can be broadcast are so noted.

A message is terminated solely by a delay of at least 3.5 character lengths at the given baud rate, and any character received after such a delay is treated as a potential address at the start of a new message.

Since only the RTU form of the protocol is being supported, each message contains binary data values and is followed by a two-byte CRC16. This is a 16-bit cyclic redundancy checksum. There are two ways to get the CRC values. One is calculated in accordance with a formula which involves recursive division of the data by a polynomial, with the input to each division being the remainder of the results of the previous one. The other is for you to use a CRC16 look-up table that is referenced in the Modicon Modbus Protocol Reference Guide. More information on how the checksum must be calculated can be found in Modicon MODBUS Protocol Reference Guide PI-MBUS-300, Rev. F.

The following abbreviations or designations are used in specifying the message formats:

| | |
|-------|--|
| IA | Instrument Address |
| FN | Function Number |
| PCH | Parameter Code High |
| PCL | Parameter Code Low |
| NBH | Number of Bits High |
| NBL | Number of Bits Low |
| NWH | Number of Words High |
| NWL | Number of Words Low |
| COUNT | Number of bytes of data |
| CRCH | CRC16 - Cyclic Redundancy Check Value High |
| CRCL | CRC16 - Cyclic Redundancy Check Value Low |

In depicting message formats, parenthesis are used to represent a byte of data. The general format of the message sent to the instrument will consist of 8 or more bytes, consisting of a "standard preamble" (IA and FN bytes), data (dependent on the function), and a CRC16 as follows:

(IA) (FN) (data) (data) (data) (data) (CRCL) (CRCH)

The MODBUS documentation refers to a Starting Address as the pointer to Coil or Input Bits, or Holding or Input Registers. They will be referred to as Parameter Codes in this document.

3.7 Read n Bits - Function Number 01/02

The message sent to the instrument will consist of 8 bytes consisting of the standard preamble, followed by the parameter code of the first data bit to be read, and the two-byte bit count to be read, as follows:

(IA) (1 or 2) (PCH) (PCL) (NBH) (NBL) (CRCL) (CRCH)

The normal reply will echo the first 2 characters of the message received, and will then contain a single-byte data byte count, which will not include itself or the CRC. For this message, there will be one byte of data per eight bits-worth of information requested, with the LSB of the first data byte transmitted depicting the state of the lowest-numbered bit parameter required.

(IA) (FN) (COUNT) (8 bits) (8 bits) (etc.) . . . (last 8 bits) (CRCL) (CRCH)

This function will be used largely to report instrument status information, such as local/remote or auto/manual status, and so a bit set to 1 indicates that the corresponding feature is currently enabled/active/true, and a bit reset to 0 indicates the opposite.

If an exact multiple of eight bits is not requested, the data is padded with trailing zeros to preserve the 8-bit format.

3.8 Read n Words - Function Number 03/04

The message sent to the instrument will consist of 8 bytes consisting of the standard preamble, followed by the parameter code of the first parameter to be read, and the two-byte word count to be read, as follows:

(IA) (3 or 4) (PCH) (PCL) (NWH) (NWL) (CRCL) (CRCH)

The reply sent by the instrument echoes the first two characters received and then contains a single-byte data byte count, the value of which does not include either itself or the CRC value to be sent. For this message, the count equals the number of words read times two. Following the byte count, the number of bytes are transmitted, MSB first, followed by the CRC16.

(IA) (FN) (COUNT) (HI) (LO) (HI) (LO) (etc.) (etc.) . . . (last HI) (last LO) (CRCL) (CRCH)

3.9 Write 1 Bit - Function Number 05

The message sent to the instrument will consist of 8 bytes consisting of the standard preamble, followed by the parameter code of the bit to set and a two-byte word whose MSB contains the desired truth value of the bit, expressed as 0xFF (TRUE) or 0x00 (FALSE), as follows:

(IA) (5) (PCH) (PCL) (State) (0) (CRCL) (CRCH)

Generally, this function will be used to control such features as auto/manual. The normal reply sent by the instrument will be a byte-for-byte echo of the message received.

(IA) (5) (PCH) (PCL) (State) (0) (CRCL) (CRCH)

3.10 Write 1 Word - Function Number 06

The message sent to the instrument will consist of 8 bytes consisting of the standard preamble, followed by the parameter code of the parameter to be written, and the two-byte value to which the parameter will be set, as follows:

(IA) (6) (PCH) (PCL) (Value HI) (Value LO) (CRCL) (CRCH)

The normal response is to echo the message in its entirety.

3.11 Write n Words - Function Number 16

The message sent to the instrument will consist of 11 or more bytes consisting of the standard preamble, followed by the parameter code of the first parameter to be written, a two-byte word count, a one-byte COUNT, and the series of two-byte words to which the parameters will be set. * See NOTE at Section 4.3 Multiple Parameter Value Communications, page 23.

(IA) (16) (PCH) (PCL) (NWH) (NWL) (COUNT) (Word HI) (Word LO)
(Word HI) (Word LO) (Word HI) (Word LO) (CRCL) (CRCH)

The instrument normally responds with a 8 byte reply, as follows:

(IA) (16) (PCH) (PCL) (NWH) (NWL) (CRCL) (CRCH)

3.12 Loopback Diagnostic Test - Function Number 08

The message sent to the instrument will consist of 8 bytes consisting of the standard preamble, followed by a four bytes of zeros, as follows:

(IA) (8) (0) (0) (0) (0) (CRCL) (CRCH)

Full MODBUS support in this area is not appropriate - consequently, the diagnostic code supported is code 0000. In response to the message the instrument must echo the message received exactly.

3.13 Error and Exception Responses

If the instrument receives a message which contains a corrupted character (parity check fail, framing error. etc.), or if the CRC16 check fails, the instrument ignores the message. If the message is otherwise syntactically flawed (e.g. the byte count or word count is incorrect) the instrument will also not reply.

If the instrument receives a syntactically correct message which contains an illegal value, it will send an exception response, consisting of five bytes as follows:

(IA) (FN) (Exception Number) (CRCL) (CRCH)

The Function Number field consists of the function number contained in the message which caused the error, with its most significant bit set (i.e. function 6 becomes x86), and the Exception Number is one of the codes contained in the following table:

| Code | Name | Cause |
|------|----------------------|---|
| 1 | ILLEGAL FUNCTION | Function number out of range |
| 2 | ILLEGAL DATA ADDRESS | Parameter ID out of range or not supported |
| 3 | ILLEGAL DATA VALUE | Attempt to write invalid data or not enough data words - action not carried out |
| 4 | DEVICE FAILURE | N/A |
| 5 | ACKNOWLEDGE | N/A |
| 6 | BUSY | N/A |
| 7 | NEGATIVE ACKNOWLEDGE | N/A |

SECTION 4.0 - PARAMETER VALUE TYPES

There are three different types of parameter values, Bit parameter, Single parameter, and Multiple parameter values.

4.1 Bit Parameter Communications

The bit parameters are fairly straight forward. The bits are accessed and transmitted per the protocol while they are stored in the instrument as one-byte entities.

The computer can read single or multiple bits using Function Numbers 01 or 02. The computer can write a single bit using Function Number 05. We do **not** support multiple bit writes via Function Number 15, as this is seldom done in real world applications and MMI or SCADA programs don't always support it either.

4.2 Single Parameter Value Communications

The two-byte integer values are straight forward. Values are stored, accessed, transmitted and treated simply as two-byte or word oriented entities.

4.3 Multiple Parameter Value Communications

In addition to being able to read or write any one of the data types in a single operation, Function Numbers 03/04 or 16 can be used to read or write a series of consecutive registers that correspond to any mix of data types, provided the number of consecutive registers falls on a parameter boundary. Otherwise, the instrument will not accept the command and will respond with an Exception Code of 3, ILLEGAL DATA VALUE. To simplify coding, the instruments will process data in order of occurrence, and only not accept/process the command for the last parameter if it does not fall on the end of the parameter boundary.

***NOTE:** Support for multi-parameter writes is limited to support of the Multi-word Write Function (Number 16), but will permit writing of one parameter only per message. The multi parameter read function supports a maximum of 10 parameters in one message.

SECTION 5.0 - EXAMPLES

Read Example #1 :

Read the process value from the instrument. Please note: you would use Modbus Function 3 (Read n Words). Abbreviated in the example as [FN].

The message sent to “**read from**” the instrument’s address 2 [IA] would be as follows:

| | [address of 1st word] | [number of words] | [CRC16] |
|-----------------|------------------------|-------------------|---------------|
| (IA) (FN) | (High byte) (Low byte) | (NBH) (NBL) | (CRCL) (CRCH) |
| Hex out: | (02) (03) (00) (01) | (00) (01) | (D5) (F9) |

The response if the process value is “79” :

| (IA) (FN) | (count) | (1st Value) | (Last Value) | (CRCL) | (CRCH) |
|----------------|-----------|-------------|--------------|--------|-----------|
| Hex in: | (02) (03) | (02) | (00) | (4F) | (BD) (BO) |

Read Example #2 :

Read the setpoint value (**Word parameter 2**) from the instrument. Please note: you would use Modbus Function 3 (Read n Words).

The message sent to “**read from**” the instrument’s address 2 [IA] would be as follows:

| | [address of 1st word] | [number of words] | [CRC16] |
|-----------------|------------------------|-------------------|---------------|
| (IA) (FN) | (High byte) (Low byte) | (NBH) (NBL) | (CRCL) (CRCH) |
| Hex out: | (02) (03) (00) (02) | (00) (01) | (25) (F9) |

The response if the setpoint value is “200” :

| (IA) (FN) | (count) | (1st Value) | (Last Value) | (CRCL) | (CRCH) |
|----------------|-----------|-------------|--------------|--------|-----------|
| Hex in: | (02) (03) | (02) | (00) | (C8) | (FD) (D2) |

Write Example #1:

The message sent to **write** “450” to the instrument’s **setpoint (Word parameter 2)** would be as follows:

| | [address of 1st word] | [value of 450] | [CRC16] |
|-----------------|------------------------|-----------------------|---------------|
| (IA) (FN) | (High byte) (Low byte) | (Value Hi) (Value Lo) | (CRCL) (CRCH) |
| Hex out: | (02) (06) (00) (02) | (01) (C2) | (A8) (38) |

The response if the setpoint value is changed to “450” and accepted is:

| (IA) (FN) | (High byte) (Low byte) | (NBH) (NBL) | (CRCL) (CRCH) |
|----------------|------------------------|-------------|---------------|
| Hex in: | (02) (06) (00) (02) | (01) (C2) | (A8) (38) |

The instrument should respond by echoing the message. If the instrument did not accept the new setpoint value, say because of a setpoint limit or out range the response would be:

| (IA) (FN)* | (Exception #) | (CRCL) | (CRCH) |
|----------------|---------------|--------|-----------|
| Hex in: | (02) (86) | (03) | (F2) (61) |

***Note: See Error and Exception Response Section 3.13, page 22.**

SECTION 6.0 - MODBUS REGISTERS

Section 6.1 - Bit Parameters

| Parameter | No. | Notes |
|----------------------------|------------|--------------|
| Communication Write Status | 1 | R/O |
| Auto / Manual | 2 | |
| Self Tune | 3 | |
| Pretune | 4 | |
| Alarm 1 Status | 5 | R/O |
| Alarm 2 Status | 6 | R/O |
| Auto Pretune | 7 | |
| Change Indicator | 8 | R/O |
| Loop Alarm Status | 9 | |

Section 6.2 - Word Parameters

| Parameter | | Notes |
|----------------------------|----|-------------------|
| Process Variable | 1 | R/O |
| Setpoint | 2 | Target if ramping |
| Output Power | 3 | |
| Deviation | 4 | R/O |
| PB2 | 5 | |
| PB1 | 6 | |
| Controller Status | 7 | R/O |
| Reset (or Loop Alarm Time) | 8 | |
| Rate | 9 | |
| Output 1 Cycle Time | 10 | |
| Scale Range Low | 11 | |
| Scale Range High | 12 | |
| Alarm 1 Value | 13 | |
| Alarm 2 Value | 14 | |
| Manual Reset | 15 | |
| Overlap / Deadband | 16 | |
| On / Off Differential | 17 | |
| Decimal Point Position | 18 | |
| Output 2 Cycle Time | 19 | |
| Output 1 Power Limit | 20 | |
| Reserved | 21 | |
| Setpoint High | 22 | |
| Setpoint Low | 23 | |
| Filter Time Constant | 24 | |
| Process Value Offset | 25 | |
| Recorder output Max | 26 | |
| Recorder output Min | 27 | |

| | | |
|--------------------|----|-----|
| Alarm 1 Hysteresis | 28 | |
| Alarm 2 Hysteresis | 29 | |
| Motor Travel Time | 30 | R/O |
| Minimum 'ON' Time | 31 | |

Section 6.2 - Parameters (cont):

| Parameter | | Notes |
|------------------------------------|-----|---|
| Segment Mode | 32 | Time or Rate Mode |
| Programmer Status 1 | 33 | R/O - As per WEST ASCII - See Note* |
| Programmer Status 2 | 34 | R/O - As per WEST ASCII - See Note* |
| Current Program Number | 35 | R/O Indicates Current Program running/in hold |
| Current Segment Number | 36 | R/O Indicates Current Segment running/in hold |
| Current Setpoint Value | 37 | R/O |
| Current Segment Event Status | 38 | R/O |
| Segment Time Remaining | 39 | R/O |
| Programmer Commands | 40 | As per WEST ASCII - See Note* |
| Delayed Start Time | 41 | |
| Power Fail Recovery | 42 | Cold or Warm Start |
| External Selection Control | 43 | As per WEST ASCII - See Note* |
| Hold Time Value | 44 | R/O |
| Program Index -used for read/write | 45 | Used with registers 1001 to 1058 |
| SP Ramp Rate (1462 only) | 46 | Setpoint Ramp Rate |
| Go To (1462 only) | 47 | Defines Base mode status at end of each program |
| Recovery Type (1462 only) | 48 | Defines power recovery type |
| Outputs (1462 only) | 49 | Set Output status in Base mode |
| Manufacturer ID | 121 | R/O -231 Representing "W1" |
| Equipment ID | 122 | R/O - number 1460 |

*Note: See page 10 for the appropriate choices.

Section 6.3 - Segment Data

| Parameter | No. | Notes |
|----------------------------------|------|-------|
| Number of Cycles Programmed | 1001 | |
| Number of Cycles Completed | 1002 | R/O |
| Preset x60 Mode | 1003 | |
| Auto-Hold Type | 1004 | |
| Hold On | 1005 | |
| Hold Band Value | 1006 | |
| Final Setpoint Value (Segment 1) | 1007 | |
| Final Setpoint Value (Segment 2) | 1008 | |
| Final Setpoint Value (Segment 3) | 1009 | |
| Final Setpoint Value (Segment 4) | 1010 | |
| Final Setpoint Value (Segment 5) | 1011 | |
| Final Setpoint Value (Segment 6) | 1012 | |
| Final Setpoint Value (Segment 7) | 1013 | |
| Final Setpoint Value (Segment 8) | 1014 | |

| | | |
|-----------------------------------|------|--|
| Final Setpoint Value (Segment 9) | 1015 | |
| Final Setpoint Value (Segment 10) | 1016 | |
| Final Setpoint Value (Segment 11) | 1017 | |
| Final Setpoint Value (Segment 12) | 1018 | |
| Final Setpoint Value (Segment 13) | 1019 | |
| Final Setpoint Value (Segment 14) | 1020 | |
| Final Setpoint Value (Segment 15) | 1021 | |

Section 6.3 - Segment Data (cont)

| | | |
|-------------------------------------|-------------|--|
| Final Setpoint Value (Segment 16) | 1022 | |
| Time/Rate Value (Segment 1) | 1023 | |
| Time/Rate Value (Segment 2) | 1024 | |
| Time/Rate Value (Segment 3) | 1025 | |
| Time/Rate Value (Segment 4) | 1026 | |
| Time/Rate Value (Segment 5) | 1027 | |
| Time/Rate Value (Segment 6) | 1028 | |
| Time/Rate Value (Segment 7) | 1029 | |
| Time/Rate Value (Segment 8) | 1030 | |
| Time/Rate Value (Segment 9) | 1031 | |
| Time/Rate Value (Segment 10) | 1032 | |
| Time/Rate Value (Segment 11) | 1033 | |
| Time/Rate Value (Segment 12) | 1034 | |
| Time/Rate Value (Segment 13) | 1035 | |
| Time/Rate Value (Segment 14) | 1036 | |
| Time/Rate Value (Segment 15) | 1037 | |
| Time/Rate Value (Segment 16) | 1038 | |
| Event Outputs (1..4) (Segment 1) | 1039 | |
| Event Outputs (1..4) (Segment 2) | 1040 | |
| Event Outputs (1..4) (Segment 3) | 1041 | |
| Event Outputs (1..4) (Segment 4) | 1042 | |
| Event Outputs (1..4) (Segment 5) | 1043 | |
| Event Outputs (1..4) (Segment 6) | 1044 | |
| Event Outputs (1..4) (Segment 7) | 1045 | |
| Event Outputs (1..4) (Segment 8) | 1046 | |
| Event Outputs (1..4) (Segment 9) | 1047 | |
| Event Outputs (1..4) (Segment 10) | 1048 | |
| Event Outputs (1..4) (Segment 11) | 1049 | |
| Event Outputs (1..4) (Segment 12) | 1050 | |
| Event Outputs (1..4) (Segment 13) | 1051 | |
| Event Outputs (1..4) (Segment 14) | 1052 | |
| Event Outputs (1..4) (Segment 15) | 1053 | |
| Event Outputs (1..4) (Segment 16) | 1054 | |
| Tag Name (1st/2nd characters) | 1055 | |
| Tag Name (3rd/4th characters) | 1056 | |
| Tag Name (5th/6th characters) | 1057 | |
| Tag Name (7th/8th characters) | 1058 | |
| Program 1 (parameter list-as above) | 1101 - 1158 | |
| Program 2 (parameter list-as above) | 1201 - 1258 | |

| | | |
|-------------------------------------|-------------|--|
| Program 3 (parameter list-as above) | 1301 - 1358 | |
| Program 4 (parameter list-as above) | 1401 - 1458 | |
| Program 5 (parameter list-as above) | 1501 - 1558 | |
| Program 6 (parameter list-as above) | 1601 - 1658 | |
| Program 7 (parameter list-as above) | 1701 - 1758 | |
| Program 8 (parameter list-as above) | 1801 - 1858 | |

***Note:** Register 1001...1058 will access the program number setup by “Parameter Number” (Register 41)